Predicting civil unrest by categorizing Dutch twitter events

van Noord, Rik; Kunneman, Florian A.; van den Bosch, Antal

Published in:
BNAIC 2016 Benelux Conference on Artificial Intelligence

DOI:
10.1007/978-3-319-67468-1_1

Publication date:
2017

Predicting civil unrest by categorizing Dutch Twitter events

Rik van Noord [a] Florian A. Kunneman [b] Antal van den Bosch [b]

[a] *Institute of Artificial Intelligence, University of Groningen*
[b] *Centre for Language Studies, Radboud University*

Abstract

We propose a system that assigns topical labels to automatically detected events in the Twitter stream. The automatic detection and labeling of events in social media streams is a 'big data' problem. The early detection of future social events, specifically those associated with civil unrest, has a wide applicability in areas such as security, e-governance, and journalism. We used machine learning algorithms and encoded the social media data using a wide range of features. Experiments show a high-precision (but low-recall) performance in the first step. We designed a second step that exploits classification probabilities, boosting the recall of our category of interest, social action events.

1 Introduction

Many instabilities across the world stem from civil unrest, often involving crowd actions such as mass demonstrations and protests. A prime example of a mass crowd action in the Netherlands was the *Project X* party in Haren, Groningen, on September 21st 2012. A public Facebook invitation to a birthday party of a 16-year old girl ultimately led to thousands of people rioting.[1] The riots could only be stopped by severe police intervention, resulting in more than 30 injuries and up to 80 arrests. Afterwards it was concluded that the police were insufficiently prepared and that they were not well enough informed about the developments on social media. An evaluation committee recommended the development of a nation-wide system able to analyze and detect these threats in advance.[2] In this paper, we describe a system that leverages posts on Twitter to automatically predict such civil unrest events before they happen.

To facilitate this objective we start from a large set of open-domain events that were automatically detected from Twitter from a period spanning multiple years by the approach described in [7]. From this set we aim to identify the events that might comprise civil unrest, henceforth *social action events*. Instead of focusing on this event type only, we categorize all events into a broad categorization of events, and distinguish social actions as one of the event types.

2 Related Work

2.1 Predicting social action events

Only a few works aim to detect social action events. [3] try to predict civil unrest in South America based on Twitter messages. In contrast to our approach, they predict such events directly from tweets, by matching them with specific civil unrest related keywords, a date mention, and one of the predefined locations of interest. Their system obtains a precision of 0.55 on a set of 283 predefined events. The main drawback of their approach is that it has no predictive abilities. For example, the system is not able to detect social action events that use newly emerging keywords for a specific event, or take place in a new location. As a consequence, their system likely has a low recall; many future social actions are likely to go undetected.

A more generic approach to detecting social action events is the EMBERS system by [13]. They try to forecast civil unrest by using a number of open source data sources such as Facebook, Twitter, blogs,

[1] http://www.nu.nl/binnenland/2915769/facebook-feest-haren-ontaardt-in-chaos-en-rellen.html
[2] http://nos.nl/artikel/482043-cohen-fouten-politie-burgemeester.html

news media, economic indicators, and even counts of requests to the TOR browser.[3] Using multiple models, the system issues a warning alert when it believes a social action event is imminent. Tested over a month, the EMBERS system obtained a precision and recall of respectively 0.69 and 0.82. In follow-up work, [10] report on the results of EMBERS when only taking Twitter information into account, mentioning a precision of 0.97 but a recall of 0.15.

2.2 Categorizing events

Some approaches based on Twitter perform some form of broad categorization (e.g. [15]; [12]). In these approaches there is no event detection procedure before dividing the data into different categories. The described approaches either identify which topics are often talked about on Twitter, or focus on the categorization of users instead of events. To our knowledge, the only approach that focuses on the categorization of automatically detected events is by [14]. They apply Latent Dirichlet Allocation [2] to a set of 65 million events to generate 100 topical labels automatically. Manual post-annotation winnowed these down to a set of 37 meaningful categories. 46.5% of the events belong to one of these categories, while 53.5% of the events are in a rest category. [14] compared their unsupervised approach to categorizing Twitter events to a supervised approach. They selected the best 500 events (detected with the highest confidence) and manually annotated them by event type. Their unsupervised approach obtained an F1-score of 0.67, outperforming the supervised approach which obtained an F1-score of 0.59. However, they do show that the F1-score of the supervised approach steadily increases when using more training instances.

3 Experimental Set-up

Our study starts with a set of automatically detected events from Twitter, described in Section 3.1.1. We manually annotate a subset of these events by type, and subsequently train a machine learning classifier on several feature types extracted from these events. Performance is both evaluated on the annotated event set and on the larger set of remaining events.

3.1 Data

3.1.1 Event set

To perform automatic event categorization, we use the event set described in [6] which was extracted based on the approach described in [7]. As this approach was applied to Dutch tweets, the set mainly comprises Dutch events. The event detection approach is based on the method of [14], who used explicit future time expressions to identify events. Each event has a set of attributes, such as the date, keywords and event score. The event score is linked to the size and popularity of an event. For the exact calculation of this score, we refer to [7, pp.13]. Over a 6-year period (2010-2015), [6] ultimately obtained 93,901 events. This event set is used for our categorization system.

3.1.2 Event annotations

We select two sets of events for categorization. Our first event set contains the 600 events with the highest event score in the output of [6]. This enables us to make an approximate comparison to [14], who evaluated their system on the basis of their best 500 events. We refer to the set of events with the highest event scores as the *best event set*.

Our second event set is created by randomly selecting an event from the ranked total event set for intervals of 155 events (with all events ranked by event score), excluding the best 600 events of the best event set. We refer to this event set as the *random event set*. Non-Dutch events were manually removed from both event sets, leaving 586 of the best events and 585 of the random events suitable for annotation.

Seven different annotators were involved in the annotation process, who all at least annotated 40 and at most 175 of the events in the best event set. 195 of the 586 best events received a double annotation so that we are able to calculate inter-annotator agreement. The other 390 events, as well as the 585 random

[3] An indication of the number of people who chose to hide their identity and location from the online community.

events, were annotated by one annotator. Similar to [14], the annotator is asked two questions for each event:

- Is this an actual event according to the definition?
- What is the category of this event?

An event in our full event set is not necessarily a proper event according to the definition, as the detection procedure makes errors. Since we are not interested in the category of a non-event, the events that are annotated as a non-event are filtered from the event set.

We defined ten possible categories after an initial manual inspection of about 200 events. They are listed in Table 1. *Social action* is the category of interest. As arguably less straightforward categories we included *special day* and *advertisement* because manual inspection of the data suggested that those types of events were frequent enough to deserve their own category.

Table 1: The ten different categories with examples.

Category	**Example event**
Sport	Soccer match, local gymnastics event
Politics	Election, public debate
Broadcast	Television show, premiere of a movie
Public event	Performance of a band, festival
Software	Release of game, release of new iPhone
Special day	Mother's Day, Christmas
Social action	Strikes, demonstrations, flashmobs
Celebrity news	Wedding or divorce of a celebrity
Advertisement	Special offers, retweet and win actions
Other	Everything that does not fit in one of the other categories

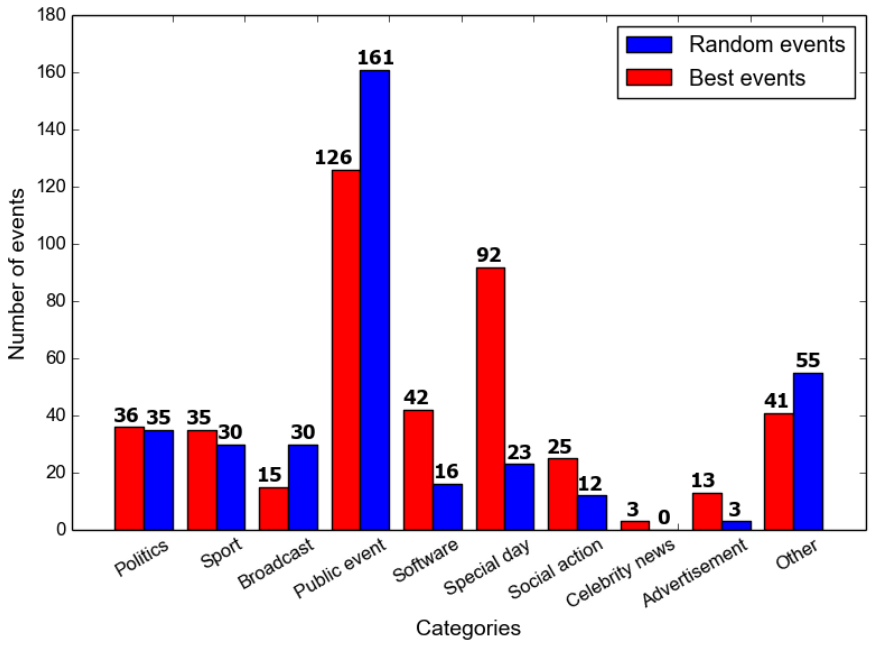

Figure 1: The ten different categories with the number of annotated examples in the best and random event set.

The 195 events annotated by two coders yielded a Krippendorff's alpha [5] of 0.81 on judging whether or not it was an actual event and 0.90 on categorizing events. These scores can be consid-

ered excellent and show that we can reliably view the events that were annotated once as if they were annotated correctly. Therefore, the 586 random events could be annotated once by two annotators.

Events that were (at least once) annotated as a non-event are removed from the event set, as well as events where annotators disagreed on the category. 27.4% of the best events were a non-event, leaving 425 of the best events. In the random event set 38.1% were discarded as non-events (leaving 362 events).

The annotations by event category are shown in Figure 1. *Public event* is the dominant category, comprising 29.6% of the best events and 44.5% of the random events. Most other event categories occur regularly, with the exception of *advertisement* and *celebrity news*. The latter category was so infrequent that it was removed from both event sets. *Advertisement* was removed from the random event set, but was retained for the best event set.

3.2 Training and testing

Based on the annotated events we trained a machine learning classifier to distinguish the ten event types. We describe the event features, classification approaches and evaluation below.

3.2.1 Feature extraction

Table 2: Types of extracted features

Tweet features	Tweet count	Bag-of-words	Sentiment		
Event features	Event-score	Keyword-score	Event date	Periodicity	Wikipedia

To enable the classifier to learn the specific properties of each event category we extract several types of features from each event. They are listed in Table 2, and can roughly be divided into 'Tweet features' and 'Event features'. Tweet features comprise characteristics of the tweets that refer to an event. We extract three types of features from them. First, the number of tweets, which might reflect the popularity of an event. We also distinguish between the numbers of tweets before, during or after[4] an event. Second, we extract each word used in the event tweets as a feature, jointly referred to as Bag-of-words features. As a third feature type we scored the average subjectivity and polarity of each event tweet, using the approach by [4]. The subjectivity and polarity score of the event are averaged over all event tweets. Some event types might be referred to fairly objectively in tweets, while others might stir more sentiment.

The first three types of Event features are derived from [7]. The first feature type, the event score, describes the link between the event keywords and the date of the event. This score gives an indication of the confidence that the set actually represents an event. Second, the keyword scores give an indication of the commonness of each event keyword, based on the commonness score as described in [8]. Third, the event date might help to recognize event types that are linked to big events, such as elections. The fourth feature type indicates whether an event is of a periodic nature. This feature is based on the output from a periodicity detection system described by [6]. Fifth, we employ DBpedia [1] in order to generalize over the different named entities present in the events. Since we want to generalize over the different terms, we are especially interested in the `type` attribute of the entity in DBpedia. This gives us a broader description of the terms than just a single value would have provided us with. For example, Feyenoord is a *SoccerClub*, *SportsTeam* and *Organisation*, while Justin Bieber is an *Artist*, *MusicalArtist*, *MusicGroup*, and a *NaturalPerson*. We extract the different DBpedia `types` for each event keyword. The keywords are linked to DBpedia using Wikification [9].

3.2.2 Classification

Based on the extracted feature sets along with the annotated categories, we train a Naive Bayes classifier[5] using the Python module Scikit-learn[6] [11]. We applied two methods to increase the performance of the

[4]Since we wanted to provide our system with as much training data as possible, we also extracted relevant tweets that were posted after the event took place. Obviously, when predicting events in the future, this type of data will be unavailable.

[5]In addition to Naive Bayes, we experimented with Support Vector Machines and K-nearest neighbors. We will only report on the outcomes of Naive Bayes, which yielded the best performance.

[6]`http://scikit-learn.org`

classifier: down-sampling the dominant *public event* class, and performing bag-of-words classification as a first-step classification. The first method simply reduces the number of *public events* in the training set to ensure it does not hinder the performance of the minority classes. The number of *public events* is reduced to the same frequency as the second-most frequent class in the event set, resulting in the deletion of 34 events in the best event set and 106 events in the random event set.

The second method only feeds the bag-of-words features to the classifier in an initial stage, and subsequently adds the resulting classification to the set of other features. The advantage of this method is that it reduces the feature space, while acknowledging the information in the word features. Also, it enables us to measure the impact of the non bag-of-words features in comparison to a bag-of-words baseline.

3.2.3 Evaluation

The performance on categorizing events is evaluated in two ways. The first is to apply 5-fold cross validation on the annotated sets of events. We do this for both the best event set (for a comparison with [14]) and the random event set, calculating the average precision, recall and F1-score.[7] The second way is to evaluate the results on a set that was never used in the training phase. The classifiers are trained on the two sets of annotated events and subsequently applied to the remaining 92,701 events. Performance on these unseen events is evaluated by manually inspecting a subset of them. In order to evaluate the precision of each classification category we randomly selected 50 events per category for evaluation, except for our category of interest, social action events, for which we include all 93 events classified with this category. *Advertisement* could only be evaluated for 25 events, as it was only predicted 25 times. This ultimately resulted in a total set of 468 events, which we refer to as the *Evaluation set*.

4 Results

4.1 Annotated set

Table 3 shows the most important results of the 5-fold cross validation. Averaged over all categories, the best event set obtained an F1-score of 0.65, while the random event set received an F1-score of 0.58. It appears to be easier to classify events with a higher event score.[8] However, we found no significant effect of event score when doing a least-squares logistic regression test for the random event set ($r(360) = -0.05$, $p = 0.39$). This suggests that there is a small subset of events with a very high event score that is easier to classify, but that there is no significant effect of event score in general.

Comparing the setting where only bag-of-words is used as a feature with the setting where the classification based on bag-of-words is added as a feature to the other features, the latter setting yields the best outcomes.

Social action is predicted at a high precision in the best events set, but the scores for the random events are poor. This might be due to the low number of instances in this set (12), in comparison with the 25 social actions in the best event set.

Table 3: The results of the 5-fold cross validation for the Naive Bayes algorithm while down-sampling the dominant *public event* class.[9]

		All categories			**Social actions**		
		Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**
Best events	Only bag-of-words	0.67	0.59	0.55	0.68	0.41	0.52
	Bag-of-words added as a feature	**0.67**	**0.67**	**0.65**	0.79	0.44	0.56
Random events	Only bag-of-words	0.61	0.60	0.57	0.40	0.17	0.24
	Bag-of-words added as a feature	**0.64**	**0.60**	**0.58**	0.40	0.17	0.24

[7] This was calculated by using the *weighted* setting in scikit-learn.

[8] Recall that the event score indicates a high confidence that the automatically detected unit is an event.

[9] Down-sampling increased the F1-score by 0.05 for the best event set and 0.06 for the random event set.

4.2 Evaluation set

Table 4 shows the results on the Evaluation set, listing the precision per category. In general, these scores are high for a 9-class classification task. The precision per class is even 1.00 for *sport* and *politics*, meaning that if the classifier predicted those categories, it did so perfectly. The categories *public event* and *advertisement* score below 0.70, however. The low precision for *public event* impacts the overall performance of the classification system substantially. As 81,538 out of 92,701 events were classified as a *public event*, a precision of 0.57 leads to about 35 thousand incorrectly classified events.

We should keep in mind that the non-events were not excluded from the full event set. It was estimated that 38.1% of all detected events are not events. In the training phase these non-events were excluded, so it is likely that the classifier will assign the non-events in the full event set to the most frequent category. A large part of the bias to *public event* may be due to the occurrence of non-events in the full event set. This leads us to conclude that if there were a more reliable way to automatically exclude non-events, the results of the general categorization would considerably improve.

The results for the *Social Action* category are promising, since the 93 *social actions* in this set were predicted with a precision of 0.80. However, we estimate that the recall of this category will be low. Only 93 out of 92,701 events (0.1%) were predicted as a *social action*, while 3.3% of events were annotated as a *social action* in the random event set.

Table 4: The precision and number of predicted instances per category.

Category	**Instances**	**Precision**	**Category**	**Instances**	**Precision**
Sport	2,771	1.00	Special day	1,722	0.78
Politics	2,170	0.86	Social action	93	0.80
Broadcast	206	1.00	Advertisement	25	0.51
Public event	81,538	0.57	Other	1,535	0.70
Software	1,630	0.96			

5 Analysis

5.1 Increasing the recall for Social Action events

Our main goal is to detect *social action events* and possibly alerting the authorities when such an event will take place. Therefore, we rather show a large list of events that might be a social action event that actually includes most of the actual events, than a system that often misses them. Since we are not talking about thousands of events daily, an analyst can annotate the set of possible social action events manually. We thus prefer a high recall to a high precision. Therefore, we propose a method to increase the recall of social action events, at minimal precision costs.

In order to increase recall we make use of the Naive Bayes classifier probability by category that is assigned to each event. Events for which *social action* obtained the second highest probability are now completely ignored, One way to remedy this is to classify all events where *social action* was the second most probable class. We refer to these events as **secondary social action events**.

By doing this we were able to expand this set with 226 additional events, which we annotated manually. 26 of the 226 *secondary social actions* were annotated as a non-event and were thus excluded from the set. 130 of the remaining 200 events were indeed annotated as a social action, resulting in a precision of 0.65. Adding the 200 events to the *social action* events in the evaluation set results in a drop of total precision from 0.80 to 0.69. However, recall was increased by **232%** while the precision only dropped by **14%**. Hence, including the *secondary social action events* seems a useful method for increasing the recall, while only mildly hurting the precision.

5.2 Most informative features

In order to achieve some insight from the most informative features for the two event sets, we calculated the chi-squared value for each feature in relation to the category label. These are listed in Table 5. The

most informative features are generally intuitive. They include words such as *stemmen (to vote)* and *stem (vote)* as indicators of a political event, but also specific hashtags such as *#VVD* and *#CDA*; CDA and VVD are political parties in The Netherlands. The best predictors for *sport* are the DBpedia type features *SoccerClub* and *ClubOrganization*. The most indicative features of the category *social action* are the words *protest* and *demonstratie (demonstration)*. Although these words almost exclusively occurred in *social action events*, due to their low frequency they do not rank in the feature top 100.

The polarity, subjectivity and periodicity features turned out to be less valuable, ranking in the bottom 25% of all features. This is surprising, since *special days* are often periodic, while it is, for example, uncommon for *social action events* to be periodic.

Table 5: The eight best features for the best and random event set, based on their chi-squared value. Non-word features are in italics. Features are only included if they occurred at least ten times in their event set.

Best events		**Random events**	
Feature	**Category**	**Feature**	**Category**
stemmen (vote)	Politics	*ClubOrganization*	Sport
stem (vote)	Politics	*SoccerClub*	Sport
19-03-2014	Politics	wint (wins)	Sport
SoccerClub	Sport	wedstrijd (match)	Sport
#vvd	Politics	2015	Politics
wedstrijd (match)	Sport	seizoen (season)	Broadcast
ClubOrganization	Sport	tv	Broadcast
#cda	Politics	tegen (against)	Sport

6 Conclusion and discussion

In this study we presented a generic event categorization system which we evaluated particularly on its ability to predict civil unrest. The general categorization system has a bias towards the dominant category *public event*, but has a high precision for the other categories, including *social action*. The recall for *social action* was low; a follow-up step that exploited the specific per-class probabilities generated by the Naive Bayes classifier led to a considerable improvement in recall of 232%, at the minor cost of a 14% decrease in precision.

The study by [14] is the only related work in the literature that also produced an extensive evaluation of event categorization, evaluating their system on a set of 500 events with the highest association (similar to the event score by which we selected a set of best events). Their 37-class approach ultimately obtained a precision, recall and F1-score of 0.85, 0.55 and 0.67. Our system offered a comparable performance: a precision, recall and F1-score of 0.67, 0.67 and 0.65.

A comparable approach to predicting civil unrest is the EMBERS system by [13]. They evaluated their system over a period of a month, resulting in a precision and recall of respectively 0.69 and 0.82. In comparison, we obtained a higher precision while our estimated recall is lower. It is interesting to note how they received this recall score. They obtained a gold standard set of *social action events* by an independent organization that had human analysts survey newspapers and other media for mentions of civil unrest; arguably a reliable way of calculating recall in the real world. Our approach is only able to recall events that were present in the set of [6]. We have not explored ways to evaluate to what extent [6] detected all *social action events* that actually happened. We should consider the possibility that we might still miss *social action events* that were never detected as events in the first place, lowering our estimated recall.

Using the ranking of the Bayesian probabilities helped to increase the recall of *social action events* by 232%. We did not use the actual probabilities to influence the classification process, but used only the ranking of these probabilities. A potential direction for future research is to use the per-class probabilities generated by the Naive Bayes classifier in a more sophisticated manner. For example, it is possible to learn a certain probability threshold for *social action* and classify events that exceed this threshold as *social action*, regardless of the probability of other categories. The actual implementation of such a

method requires a search for the best threshold setting. The main advantage of this approach is that this allows us to specify a specific precision-recall trade-off that is the most suitable for predicting social action events.

References

[1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165, 2009.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[3] R. Compton, C.-K. Lee, T.-C. Lu, L. de Silva, and M. Macy. Detecting future social unrest in unprocessed twitter data:emerging phenomena and big data. In *Intelligence and Security Informatics (ISI), 2013 IEEE International Conference On*, pages 56–60. IEEE, 2013.

[4] T. De Smedt and W. Daelemans. Pattern for python. *The Journal of Machine Learning Research*, 13(1):2063–2067, 2012.

[5] A. F. Hayes and K. Krippendorff. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89, 2007.

[6] F. Kunneman and A. van den Bosch. Automatically identifying periodic social events from twitter. *Proceedings of the RANLP 2015*, pages 320–328, 2015.

[7] F. Kunneman and A. van den Bosch. Open-domain extraction of future events from twitter. *Natural Language Engineering*, 2016.

[8] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 563–572. ACM, 2012.

[9] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.

[10] S. Muthiah, B. Huang, J. Arredondo, D. Mares, L. Getoor, G. Katz, and N. Ramakrishnan. Planned protest modeling in news and social media. In *AAAI*, pages 3920–3927, 2015.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] D. Ramage, S. T. Dumais, and D. J. Liebling. Characterizing microblogs with topic models. *ICWSM*, 10:1–1, 2010.

[13] N. Ramakrishnan, P. Butler, S. Muthiah, N. Self, R. Khandpur, P. Saraf, W. Wang, J. Cadena, A. Vullikanti, G. Korkmaz, et al. 'beating the news' with embers: Forecasting civil unrest using open source indicators. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1799–1808. ACM, 2014.

[14] A. Ritter, O. Etzioni, S. Clark, et al. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM, 2012.

[15] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, pages 338–349. Springer, 2011.